



# IALA GUIDELINE

## G1128 THE SPECIFICATION OF e-NAVIGATION TECHNICAL SERVICES

**Edition 1.7**

June 2025

urn:mrn:iala:pub:g1128:ed1.7



10, rue des Gaudines - 78100 Saint Germain en Laye, France

Tel. +33 (0)1 34 51 70 01 - [contact@iala.int](mailto:contact@iala.int)

[www.iala.int](http://www.iala.int)

International Organization for Marine Aids to Navigation

# DOCUMENT REVISION

Revisions to this document are to be noted in the table prior to the issue of a revised document.

Date	Details	Approval
December 2017	First edition	Council 65
December 2018	Edition 1.1. Update of minor issues, redrawing of figures: All drawings are remade using Enterprise Architect Figure 6: Added [0..1] to CoverageArea => geometryAsWKT <ul style="list-style-type: none"><li>This corrects an error in the previous version</li></ul> Figure 6: Changed URL to Endpoint <ul style="list-style-type: none"><li>This change is in order to embrace services that are not web-services</li></ul> Table 3: Changed Row "URL" to "Endpoint", slightly adjustment of the description text	Council 68
June 2021	Edition 1.2 Editorial corrections	
December 2021	Edition 1.3 Update of minor issues; <ul style="list-style-type: none"><li>Section 2: Added a description of the different stakeholders in Figure 1, moved explicit mentions of the MCP to informative references.</li><li>Section 3: Introduced MRNs for Service Identifiers and Organization IDs, fixed a non-existent reference to S100_Parameter by referencing S100_FC_NamedType</li><li>Section 4: Introduced MRNs for Service Identifiers and Organization IDs</li><li>Section 5: Extended the list of possible contents for the "Service Implementation and Instance Details" to composition of services and internal service structure, introduced MRNs for Service Identifiers and Organization IDs, added guidance for using keywords, defined the format of IMO and MMSI numbers in the instance description, changed the serviceType attribute to an open enumeration (including Maritime Services) and changed the multiplicity to 1..*, revision of incorrect definitions.</li></ul>	Council 74
July 2022	Edition 1.4 Editorial corrections	
June 2024	Edition 1.5 Restructuring of what information is contained in which specification level. <ul style="list-style-type: none"><li>Removed interfaces from Service Specification as interfaces cannot be technology agnostic</li></ul>	Council 80



# DOCUMENT REVISION

---

Date	Details	Approval
	<ul style="list-style-type: none"><li>• Redefined instance XML to include all data contained in other metadata.</li><li>• Removed all appendixes to come in separate files.</li><li>• Redesigned content of templates.</li></ul>	
December 2024	Edition 1.6 Consistency and terminology corrections. <ul style="list-style-type: none"><li>• Service Design document mandatory</li><li>• New Appendix D (Service Design Template for SECOM Service)</li></ul>	Transition Council 3
June 2025	Edition 1.7 Added statusEndpoint to instance metadata	Council 02

# CONTENTS

---

<b>1. INTRODUCTION .....</b>	<b>6</b>
1.1. Purpose of the Document.....	6
1.2. Link to S-100 and Product Specifications.....	6
<b>2. OVERVIEW.....</b>	<b>7</b>
2.1. Service Management Overview .....	7
2.2. Process Considerations.....	11
2.2.1. Top-Down Service Development .....	11
2.2.2. Bottom-Up Service Documentation.....	12
<b>3. SERVICE SPECIFICATION .....</b>	<b>12</b>
3.1. Service Specification .....	13
3.2. Service Specification Template .....	13
3.2.1. Introduction .....	13
3.2.2. Service Identification .....	13
3.2.3. Operational Context .....	13
3.2.4. Service Overview .....	14
3.2.5. Service Logical Data Model.....	15
3.2.6. Service Dynamic Behaviour .....	15
3.2.7. References .....	16
<b>4. SERVICE DESIGN .....</b>	<b>16</b>
4.1. Service Design .....	16
4.2. Service Design Template.....	16
4.2.1. Introduction.....	16
4.2.2. Service Design Identification .....	16
4.2.3. Technology Introduction .....	17
4.2.4. Service Design Overview.....	17
4.2.5. Physical Data Model .....	18
4.2.6. Service Interface Design .....	18
4.2.7. Service Dynamic Behaviour .....	19
4.2.8. References .....	19
<b>5. SERVICE INSTANCE DESCRIPTION .....</b>	<b>19</b>
5.1. Service Instance Description.....	19
5.2. Service Instance Description Template .....	20
5.2.1. Introduction.....	20
5.2.2. Service Instance Identification.....	20
5.2.3. Service Implementation and Instance Details .....	20
5.2.4. Release Notes .....	20
5.2.5. References .....	20
5.3. Service Instance Description XSD Structure .....	21

# CONTENTS

---

6. GOVERNANCE.....	27
7. DEFINITIONS.....	27
8. ABBREVIATIONS .....	29
9. REFERENCES .....	30
ANNEX A SERVICE SPECIFICATION TEMPLATE .....	31
ANNEX B SERVICE DESIGN TEMPLATE.....	31
ANNEX C SERVICE INSTANCE DESCRIPTION TEMPLATE.....	31
ANNEX D SERVICE DESIGN TEMPLATE FOR SECOM SERVICE .....	31

## List of Tables

Table 1	Type Definition.....	6
Table 2	Information Elements of the Service Instance Description.....	22

## List of Figures

Figure 1	Service Management Concept.....	7
Figure 2	Distinction between Service Specification, Service Design and Service Instance Description .....	9
Figure 3	Service Documentation Overview.....	9
Figure 4	Structure of the Service Instance Description.....	21



## 1. INTRODUCTION

### 1.1. PURPOSE OF THE DOCUMENT

This Guideline provides information on how to make specifications of e-Navigation Technical Services. A Maritime Service (MS) can be implemented by one or more e-Navigation Technical Services.

Taken from the concepts of service-oriented architectures, a Technical Service refers to a set of related software functionalities that can be reused for different purposes, together with policies that govern and control its usage. A Technical Service is a digital service offered by an electronic device to another electronic device.

An e-Navigation Technical Service shall be formally specified and documented as described by this Guideline. This Guideline aims to improve the visibility and accessibility of available e-Navigation Technical Services and information provided by them. This enables service providers, consumers, and regulatory authorities to share a common understanding of a Technical Service and how to implement and use it.

This Guideline is intended for service architects, system engineers and developers in charge of designing and developing a Technical Service or designing and developing a device to use it.

Furthermore, this Guideline is intended to be read by enterprise architects, service architects, information architects, system engineers and developers in pursuing architecting, design and development activities of other related services.

This document provides meta-information explaining how Technical Services shall be specified. The guidance included helps provide Service Specifications, Service Designs and Service Instance Descriptions for any kind of e-Navigation Technical Service in a standardised way. Any Technical Service Specification in the context of e-Navigation shall comply with this Guideline.

### 1.2. LINK TO S-100 AND PRODUCT SPECIFICATIONS

This Guideline is intended for, but is not limited to, Technical Services based on S-100 Product Specifications. However, since e-Navigation is currently in a transitional phase, an S-100 Product Specification may not always exist for a particular application. In that case, the proposed specification or data model shall make use of the concepts of S-100 whenever possible.

Technical Service Specifications following this Guideline shall provide references to the appropriate S-100 Product Specifications if available or to the appropriate S-100 features and attributes in their data models.

Similar to the S-100, the graphics and textual descriptions may use different data types than the XSD schemas. This is because the XSD technology does not provide these data types, but can be used synonymously.

*Table 1 Type Definition*

Type Name	Description
CharacterString (String)	A CharacterString is an arbitrary-length sequence of characters, including accents and special characters from the repertoire of one of the adopted character sets.
Real (float)	A signed real (floating point) number consisting of a mantissa and an exponent, the representation of a real is encapsulation and usage dependent.
Boolean	A value representing binary logic. The value can be either true or false.

## 2. OVERVIEW

## 2.1. SERVICE MANAGEMENT OVERVIEW

A service management concept can be visualised, as shown in Figure 1 Service Specification, product specification and Service Design documents with all their applicable accompanying files (e.g. XML schemas, OpenAPI definitions, etc) must be published at a publicly available URL defined by the approving organization and findable via MRN or URL.

Service Instance metadata shall be published in a service registry and provide a public location where the OpenAPI or similar interface definition is accessible.

In Figure 1, four different roles are envisioned in managing the contents of a service publication site and registry following this guideline.

- Service Specification producers might represent organizations governing a certain type of service type like Navigational Warning Service, Route Optimization Service, Search and Rescue Service, etc.
- Service implementers typically develop services in accordance with published specifications resulting in Service Designs. i.e. realizing a Service Specification by use of a certain technology like REST, SOAP, FTP etc.
- Service providers publish implemented technical services following a Service Design. Service providers are responsible for publishing Service Instance Descriptions ready for consumption in a service registry. Examples of service providers are shipping companies, VTS, National Maritime Administrations, etc.
- Service consumers discover service instances for consumption. In this context, the service consumer could be an actual person or an application, like an ECDIS software or perhaps a VTS software. In discovering service instances, a service consumer might need to make use of all three service description artefacts (specification, design and instance) in order to determine the full specification of the published service instance.

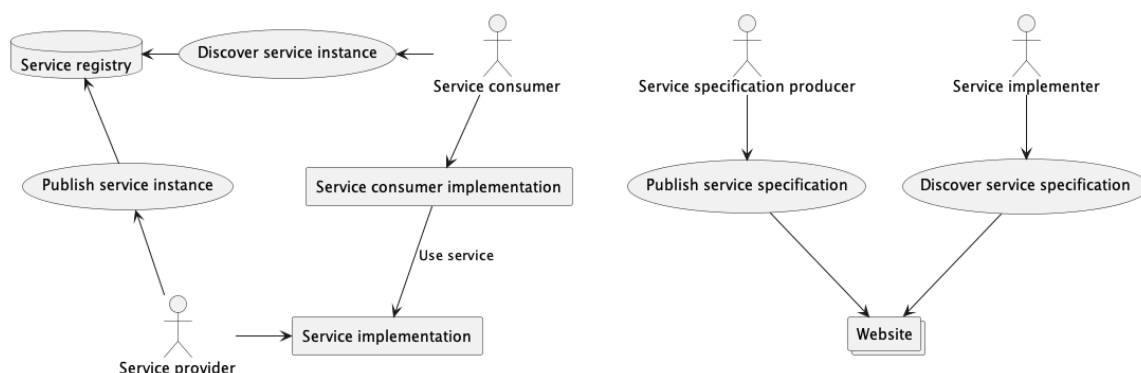


Figure 1 Service Management Concept

Figure 2 illustrates the distinction between Service Specification, Service Design and Service Instance. The Service Specification describes one dedicated service at a logical level in a technology-agnostic manner by providing, for example:



- The operational context of the service (e.g., requirements, use cases).
- The logical operations and the parameters required with respect to the logical data model (e.g. if a get-type query is an operation that is expected, what kinds of query parameters will it allow and what is expected as a return value). The logical data model used by the service (preferably defining the subset of S-100 series data that is used by the business requirements of the service).
- The dynamic behaviour of the service in the context of the use cases (sequence of logical operations).
- Author of the Service Specification (organization, contact person).

The Service Specification shall not describe the details of specific service implementation. For that purpose, a Service Design must be provided, where the actual realization of the service with a dedicated technology shall be described. The dynamic behaviour of the service must be specified in the Service Specification at the level of logical operations without tying the implementation to a given technology or architecture.

It is possible to provide different Service Designs (by using the same or different technologies), all being compliant with the same Service Specification. It is also possible to provide one Service Design that conforms to several Service Specifications, e.g., to allow backwards compatibility to older versions of a certain specification.

Each Service Design shall be documented by providing, for example:

- Reference to the Service Specification.
- Description of the chosen technology.
- The service interface descriptions (operations, parameters):
  - When using a standardized communication pattern (e.g. SECOM) the interface description may gloss over some of the operations and parameters that are defined in the standard and focus on those interfaces and parameters that have actual business case value for the use cases the service is designed to solve.
  - It is important to document how the actual interfaces correspond to the logical operations described in the specification.
- Detailed description of the used data structures (service physical data model).
- Mapping of the used data structures to the Service Specification's service data model.
- Author of the Service Design (organization, contact person).

A Service Instance (implemented according to a given Service Design) may be deployed at different locations by different service providers. For each such Service Instance, a Service Instance Description shall be provided.

Each Service Instance Description shall be documented by providing, for example:

- Reference to the Service Design (and thus, implicitly, to the Service Specification).
- Information about the service provider.
- Coverage information.



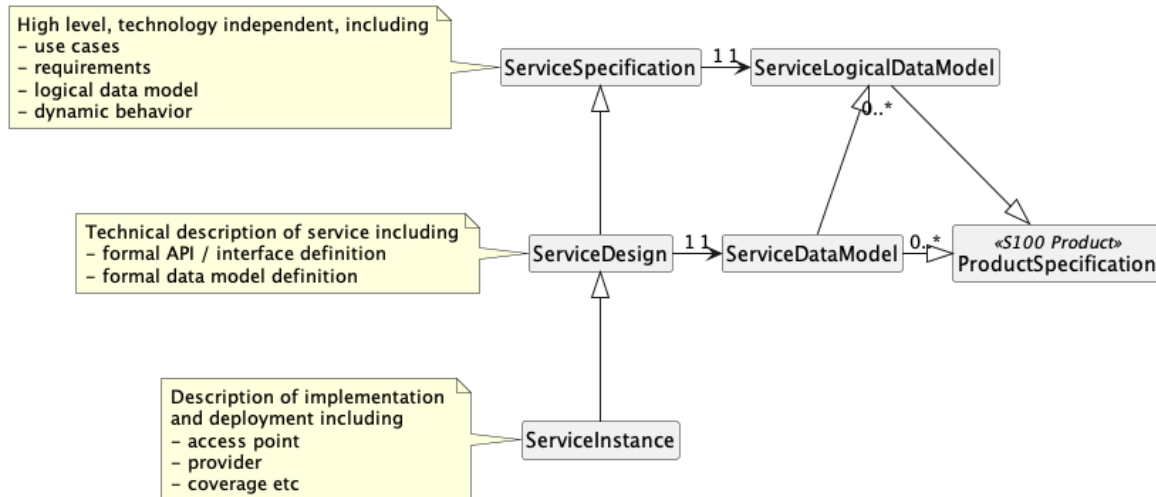


Figure 2 Distinction between Service Specification, Service Design and Service Instance Description

Figure 3 provides an overview of the Technical Service Specification artefacts. The meta-information (guidelines, templates and schemas) consists of:

- Technical Service Specification guideline – this document.
- Service Specification template – a Word document providing the framework for a textual description of a Service Specification (ANNEX A).
- Service Design template – a Word document providing the framework for a textual description of the service design (ANNEX B); Service Design templates based on a specific technology may also be made available to ease the writing and development of designs.
- Service Instance Description XSD – an XML schema definition for the formal description of a Service Instance.
- Service Instance Description template – a Word document providing the framework for a textual description of the Service Instance (ANNEX C).
- Product specification – a requirement of creating S100 products and most likely referenced as the source of the data model (both logical and physical). A service may refer to several product specifications. The requirements of product specifications are defined in [2].

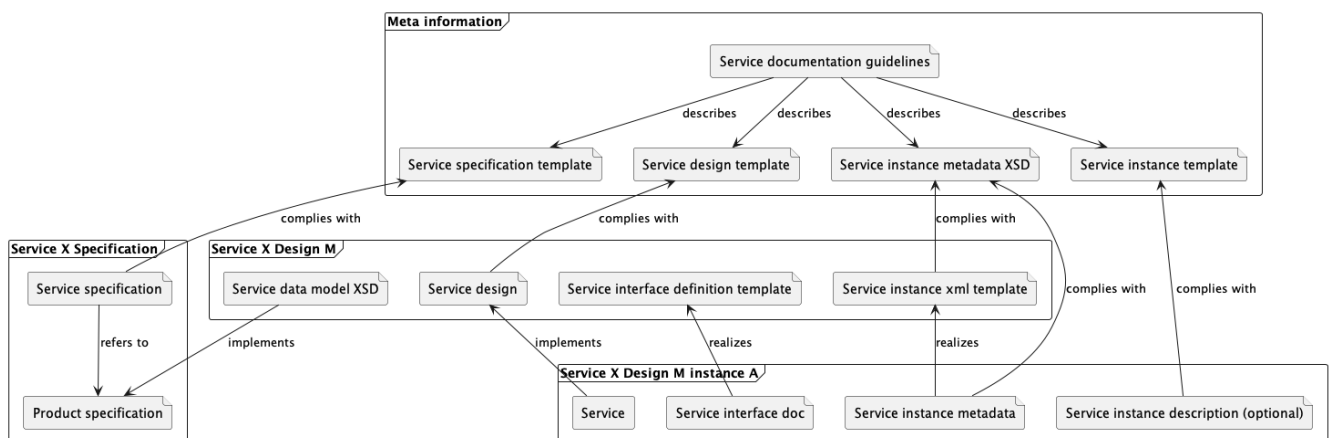


Figure 3 Service Documentation Overview



The Service Specification describes the “What”-aspects of a service, e.g., what are the characteristics of a Weather Forecast Service. This Service Specification consists of a:

- Service Specification document – a Word document (complying with the Service Specification template ANNEX A) detailing the Service Specification and focusing on the definition of the use cases, logical operations and logical data model of the service. The Service Specification must provide sufficient detail in the use cases and logical data model that the Service Designs are only an additional layer of specification without any duplication, and focus on detailing how to implement the use cases and logical data model in an actual implementation using a chosen technological approach.

The service design comprises the “How”-aspects of a service, e.g., how in detail a weather forecast service instance is implemented and accessible. Several different technical designs may exist at the same time for one Service Specification. A Service Design consists of:

- Service Design description – a document describing the Service Design.  
The Service Design description shall be using the Service Design description template in ANNEX B. If provided, the document shall refer to the Service Specification document.
- Technology dependent formal description – additional service description files as appropriate for the chosen technology (e.g., WSDL, XSD, YAML, JSON, etc.), describing the details (syntax, protocol, etc.) of the exchanged data and the interfaces supplied or required.  
The contents of this formal description must be mapped to the service data model and the logical operations defined in the specification. The means of how such mapping has to be performed is not prescribed, as they depend heavily on the chosen technology and Service Design. In some cases, the mapping is implicitly given (e.g., if the Service Design re-uses the service data model in a 1:1 manner). In other cases, a mapping table may be provided (e.g., as part of the Service Design description document), mapping each single data element of the service instance to a corresponding data element of the service data model.
- Template of the instance specification XML with all the relevant interface specification and data models defined in a formal way to reduce duplication of work from the parties implementing an instance of the service.

The lower part of the Figure 3 contains the artefacts for a dedicated instance (implementation) of a service. The service instance comprises the “Where”-aspects of a service, e.g., the actual access address (endpoint) of a weather forecast service and the geographical coverage of it. Several service instances may exist at the same time, all implementing the same Service Design. A service instance consists of:

- Deployed service provider software implementation – the actual service implementation.  
This is not part of the description, but it is the ‘subject’ that shall be described. It consists of all the software and configuration artefacts needed for providing the service.
- Service Instance Description: a document describing the actual service implementation and instantiation.  
The document shall refer to the service design description document.
- Service Instance Description XML: an XML file (following the Service Instance Description XSD), describing the service instance in a formal manner, e.g., by providing information needed for registration in the service registry. This should be an easy edit of the template provided by the Service Design.



**Note:** One service implementation (the same software) may be deployed several times at different access points. In this case, several Service Instance Description XML files will need to be produced – one for each deployed instance.

## 2.2. PROCESS CONSIDERATIONS

This document describes how the service documentation shall look like in the context of e-Navigation. Intentionally, however, this document does not prescribe any process to be followed when generating such documentation. This document does not identify any governance rules for service design and implementation (see also section 6).

This means that sub-sections 2.2.1 and 2.2.2 shall be just seen as proposals, not meaning that these are the only valid approaches. This section of the guideline is intended as a description of the different ways specifications and documentation can be created during the process of software development in the context of creating maritime technical services that are discoverable and cover a wide range of use cases that make the services applicable across the globe.

### 2.2.1. TOP-DOWN SERVICE DEVELOPMENT

In a top-down approach, the necessity of a new service and its basic outline would be first identified and described in an operational requirements document. This step is optional and out of the scope of this service documentation guidelines document.

Once the decision for building a service has been taken, a service architect (in the role of a Service Specification producer) creates the Service Specification by producing the Service Specification document and including the logical operations and service data model. If an operational requirements document exists, the Service Specification refers to it; otherwise, the requirements are documented in the Service Specification.

As soon as the Service Specification has reached sufficient maturity, it is published in an appropriate registry, most likely the envisioned IALA Maritime Resource Registry (MRR) with the unique MRN obtained from the approving body of the specification.

An interested service provider or implementer (there could also be more than one) takes the Service Specification and elaborates a Service Design for it. During this step, technology decisions are taken and documented in the service design. The service design may already be published in the appropriate registry before the service is implemented and deployed. This is useful for developers of service consumer software, as they can already base their development on the Service Design description while the service provider software is still under development.

Having the service design in place, service implementers will develop the software required for service provision.

When the service software is sufficiently mature, the service provider deploys it and publishes the access information (endpoint) and coverage area in the Service Instance Description in the service registry.

Interested service consumers can obtain Service Specifications and service design from the appropriate registry and build the required client software.

Interested service consumers with existing clients look up the service registry for service instances (complying with their choice of technology) to get the access points for the provided services in their respective geographical areas.



### 2.2.2. BOTTOM-UP SERVICE DOCUMENTATION

Existing services may be documented in a bottom-up approach. For the service to be published in the service registry, all of the following information shall be provided: a Service Instance Description, a Service Design and a Service Specification.

Assuming the service already exists, it should be easy to provide the Service Design and Service Instance Description in the structure/format described in this guidelines document. In this case, it is assumed that the technology specific data model already exists and can be directly taken as part of the Service Design description.

The service data model (part of the Service Specification) can be derived by abstracting the existing data model from technology-specific details.

The rest of the Service Specifications (interface and operation descriptions as well as requirements) shall be newly created.

Once the Service Instance Description, the Service Design and the Service Specification are sufficiently mature, they shall be published in the applicable service registries. Interested consumers may then look up services in the registries.

## 3. SERVICE SPECIFICATION

The purpose of the Service Specification is to collect the results of service identification and Service Design activities. The aim is to document the key aspects of a dedicated service at the logical level:

- The operational and business context of the service:
  - Requirements for the service (e.g., information exchange requirements).
  - Involved nodes: which operational components provide/consume the service.
  - Operational activities supported by the service.
  - Relation of the service to other services.
- The service description:
  - Logical operations of the service.
  - Service payload definition at a logical level.
  - Service dynamic behaviour description at the level of actors and use case-derived operations without technical details that tie the implementation to a certain technical design.
- Service provision and validation aspects.

The purpose of the Service Specification is to provide a holistic overview of a service and its building blocks at the logical level. The Service Specification consists of:

- (Mandatory) a Service Specification document – a human-readable documentation of the service key aspects.
- (Recommended) a model-based description – e.g., a UML model describing the logical operations and data structures.

The Service Specification document may and should re-use artefacts that were produced in the model-based description.



**Note:** The Service Specification is intended to be technology-agnostic. The Service Specification shall not describe the details of a specific service implementation. For that purpose, a Service Design shall also be provided, where the actual realisation of the service with a dedicated technology shall be described. This is the reason why the Service Specification must not include interface definitions or a formal (technical) definition of the service data model, as they are dependent on the logical data model as well as the implementation technology.

### 3.1. SERVICE SPECIFICATION

---

The purpose of the Service Specification is to document in human readable manner all the information comprising a Service Specification. A Service Specification describes one dedicated version of one dedicated service in detail at the logical level.

The Service Specification describes a well-defined baseline of the service and clearly identifies the service version. In this way, it supports the configuration management process.

The Service Specification also provides the foundation material for the future standardisation process.

A template is available to ensure a certain uniformity of the Service Specification produced by different authors.

### 3.2. SERVICE SPECIFICATION TEMPLATE

---

The Service Specification template ANNEX A shall support the service architects in creating a document-based description of the services at a high level of abstraction. The template prescribes a structure of sections (to be completed by the author of the Service Specification), and for each section, descriptive instructions for the intended content.

#### 3.2.1. INTRODUCTION

The introduction section contains the usual basic information, such as the purpose of the document, intended readership, etc.

#### 3.2.2. SERVICE IDENTIFICATION

The service identification section provides a tabular overview of mainly administrative attributes needed for the identification and lookup of the service. This must include:

- Name
- Identifier
- Version
- Author
- Keywords
- Service types of the Service Specification.

The service identifier shall be in the form of an MRN [3] (Maritime Resource Name).

#### 3.2.3. OPERATIONAL CONTEXT

The operational context section describes the context of the service from an operational perspective.

The operational context description shall be based on the description of the operational model, consisting of a structure of operational nodes.



If the service is part of one or more Maritime Services, they shall be referenced in this section.

Optionally, a simple high-level use case, described in layman's terms, could be provided as an introduction to this section.

The operational context shall be a description of how the service supports interaction among operational nodes. This can be achieved in two different levels of granularity:

- A description of how the service supports the interaction between operational nodes.

This basically consists of an overview about which operational nodes shall provide the service and which operational nodes will consume the service.

- A more detailed description stating what operational activities the service supports in a process model.

Moreover, the operational context shall describe any requirement the service shall fulfil or adhere to. This refers to functional as well as non-functional requirements at a high level (business/regulatory requirements, system requirements, user requirements). Information exchange requirements are of great interest since the major objective of services is to support interaction between operational nodes.

The source material for the operational context description should ideally be provided by operational users and is usually expressed in dedicated requirements documentation. Any applicable documents shall be mentioned in the References section. If no requirements documents are available, then the basic requirements for the service shall be defined in the Service Specification document in tabular form.

The service shall be linked to at least one requirement.

#### **3.2.4. SERVICE OVERVIEW**

This section aims to provide an overview of the main elements of the service. The elements in this view are all usually created by means of a modelling tool and should focus on logical components and actors in the use case of the service.

Architectural elements applicable for this description are:

- Service – the element representing the service, as a whole.
- Service Operations – describe the logical operations used to access the service. Actual technical designs may rename some of the operations or ignore some of the operations if the operation is triggered in a different way, e.g. API-based designs require more operations than an event-based design. Multiple service operations may also be implemented by a single API endpoint, with e.g. different HTTP methods differentiating between operations.
- Service Operations Parameter Definitions – identify logical data structures being exchanged via Service Operations. These parameters must not be an exhaustive list, but describe the parameters that are required for the business logic to work.

These elements may be depicted in one or several diagrams. Which and how many diagrams are needed depend on the chosen architecture description framework and complexity of the service.

The service overview may be described by using a suitable diagram that illustrates the logical operations and their interaction between the service provider, consumer, and other necessary parties. This information shall also be provided in text to ensure accessibility and further understanding. The text may be in tabular form or free-form text.

Using, for example, the level 1 of the C4 model [6] where the primary element is the software system in scope, supported by people and software systems that are directly connected to the software system in scope. The focus should be on people and software systems, and not on technologies, protocols, etc. The intended audience of these diagrams is both technical and non-technical people.



It is also recommended to describe the considerations resulting in the selection of a logical Message Exchange Pattern (MEP) for the logical operations. However, it is important to note that the logical operations do not need to map one-to-one to actual interfaces in the design.

### 3.2.5. SERVICE LOGICAL DATA MODEL

This section describes the data structures to be exchanged between service providers and consumers. The data model shall provide enough information allowing the implementation of the service based on this information, but on the other hand, it should describe the data structures sufficiently abstract; this means it should avoid listing all details or defining technology-specific data types. It is recommended to visualize the data structures by means of suitable diagrams. The complete information model (in the form of logical data structures) shall be shown using diagram(s) and explanatory text, e.g. tables. A description shall be given to each entity item, its attributes and the relations between entity items after each diagram that shows data items. If the data model is implemented with an S-100 product, the data model structure should take into account efficient implementation in the actual XML format and prioritise that.

If the service logical data model is related to an external data model (e.g., being a subset of a standard data model, e.g., typically based on an S-100 specification), then the service data model shall refer to it. Each data item of the service data model shall be mapped to a data item defined in the external data model. This mapping may be added in the same table that describes the data items and their attributes and associations. The idea is that, when reading the Service Specification (including the logical service data model), the reader must clearly understand the payload structures. If the service re-uses structures of an external data model, then these structures can be referred to rather than replicated in the Service Specification. The tabular presentation of the payload allows the provision of references to an externally defined model.

It is important to note that at this phase of the design of logical data models, the actual structure and format of the data must be considered and taken into account. The specification should also consider the efficiency of data transfer and not require unnecessary transfer of data.

However, while references to external data models are allowed and encouraged, all of the fields and enumerations that are used should be described in relation to the business logic of the service and make it implicit how an implementing service should use different fields and enumeration values.

In addition to the data model exchanged between service providers and consumers, this section may optionally also contain a description of the internal data model, as it seems appropriate to the service provider and/or the service consumer side. Such a description might be helpful for understanding, as it provides additional information on how the service might be built. However, this internal service data model shall be declared as informative only – it is not an authoritative part of the Service Specification.

### 3.2.6. SERVICE DYNAMIC BEHAVIOUR

This section describes the interactive behaviour between logical service components, including users, consumers, providers, and other services. The actual interfaces, technical state machines, and technical orchestration are not of specific interest at this level. The focus must be on documenting the logical sequences between actors, human or software.

The following types of views and diagrams can be used to describe the dynamic behaviour:

- activity diagrams;
- sequence diagrams; and
- other interaction diagrams.

The dynamic service behaviour description should be at the level of user stories and should cover a single user story per sub-section. Related user stories should be grouped close to each other.





This section should also include the definition of the logical operations and their expected parameters and types of return data in relation to the logical data model to better formalize the required behaviour of the service while staying technology agnostic.

### **3.2.7. REFERENCES**

The References section contains a list of all documents referred to by the Service Specification (e.g., requirements documents (if any)).

## **4. SERVICE DESIGN**

---

### **4.1. SERVICE DESIGN**

---

The purpose of the Service Design is to document in a human-readable manner all the information comprising the technology-specific design of a service. This document shall provide a detailed description of how a service shall be realised with a certain technology, including the communication pattern selected. For the technology-independent information, this document shall refer to the Service Specification document, rather than replicating any information.

Note: In theory, one service design may describe several different kinds of services. In this case, all Service Specifications shall be referenced in the Service Design. On the other hand, it is obvious that one Service Specification may be referenced by several different Service Designs. This is the case when a service shall be implemented/provided by using different technologies.

To ensure a certain uniformity of service designs produced by different authors, the document shall be aligned with the Service Design template.

Many of the currently developed service designs are made based on the SECOM standard, and therefore, a specific Service design template has been made to ease the creation of new service designs based on SECOM.

### **4.2. SERVICE DESIGN TEMPLATE**

---

The Service Design template ANNEX B shall support the service architects/designers in creating a document-based description of the service design. The template prescribes a structure of sections (to be completed by the author of the service design), and for each section, descriptive instructions for the intended content.

#### **4.2.1. INTRODUCTION**

The introduction section contains basic information, such as the purpose of the document, intended readership, etc.

#### **4.2.2. SERVICE DESIGN IDENTIFICATION**

The Service Design identification section provides a tabular overview of mainly administrative attributes needed for the identification and lookup of the Service Design.

This must include:

- Name
- Identifier
- Link to Service Specification (that design refers to)
- Version





- Author
- Vendor (optional)
- Keywords
- Service types of the Service Design.

#### 4.2.3. TECHNOLOGY INTRODUCTION

The technology introduction section contains basic background about the chosen technology. In most cases, this will be a short description of basic technology aspects accompanied by appropriate references to standards documents and best practice descriptions.

#### 4.2.4. SERVICE DESIGN OVERVIEW

This section aims to provide an overview of the main elements of the Service Design and map the design elements to the Service Specification elements. The overview should include necessary diagrams and accompanying explanatory text.

Architectural elements applicable for this description are:

- Service – the element representing the whole service.
- Service interfaces – the communication mechanisms of the service, i.e., interaction mechanisms between the service provider and the service consumer. Defined by allocating service operations to either the provider or the consumer of the service.
- Service operations – describe the operations used to access the service.
- Service operations parameter definitions – identify data structures being exchanged via Service Operations.

The above elements may be depicted in one or several diagrams. Which and how many diagrams are needed depend on the chosen architecture description framework, the chosen technology, and the complexity of the service. The diagrams should aim to cover what is described in level 2 of the C4 model [6] where the primary elements are containers (web apps, services, database schemas, etc) of the software system scope supported by people and software directly connected to the containers. The intended audience for these diagrams is technical people.

If the structure of the Service Design largely follows the Service Specification, then it is not necessary to replicate identical diagrams here in this section; in this case, this section shall contain references to the Service Specification document. However, it is assumed that in many cases, depending on the chosen technology, the actual interface and/or operation names (and structuring) are not fully identical to the abstract definition given in the Service Specification.

The Service Design overview may be described by using a suitable diagram that illustrates the service interfaces with their operations and their allocation to the service provider and the service consumer. This information shall also be provided in tabular form.

Furthermore, it shall be described how the Message Exchange Patterns (MEP) are realized with the chosen technology.

A service interface supports one or several service operations. Depending on the Message Exchange Pattern, service operations are either to be implemented by the service provider (e.g., in a Request/Response MEP, query operations are provided by the service provider – the service consumer uses them in order to submit query requests to the service provider), or by the service consumer (e.g., in a Publish/Subscribe MEP, publication operations are provided by the service consumer – the service provider uses them to submit publications to the service consumer). This distinction shall be clearly visualized. For each service interface,



it shall be stated whether it is provided or requested by the service. A service provides at least one service interface.

A paragraph detailing how the service can be discovered (e.g. by using a service registry) should be included in this section.

#### **4.2.5. PHYSICAL DATA MODEL**

This section provides a detailed description of the data structures exchanged between the service provider and the service consumer. This description shall also include a mapping of the data structures to the service data model provided in the Service Specification.

The Service Design template does not prescribe a detailed format for this section. Allowed presentations of the physical data model include:

- Diagrams that represent the data structures, including detailed physical data type descriptions at the attribute level.
- A file describing the data structures (e.g., XML/XSD, JSON). This is required when data structures are not covered by existing standards (e.g. S-100 series, SECOM, MMS).
- Tabular presentations if not covered by existing standards and their documentation.

Any mixture of the above formats is allowed. An S-100 compliant specification shall refer to the Dataset Discovery Metadata in order to link to product specifications and S-100 compliant data formats.

If the physical service data model is related to an external data model (e.g., being a subset of a standard data model, i.e., based on an S-100 specification), then this section shall refer to it; each data item of the physical data model shall be mapped to a data item defined in the external data model. This mapping may be added in the same table that describes the data items and their attributes and associations.

#### **4.2.6. SERVICE INTERFACE DESIGN**

Architectural elements applicable for this description are:

- Service interfaces.
- Service operations – functions or procedures which enable programmatic communication with a Service via a Service interface.
- Parameters – constants or variables passed into or out of a service interface as part of the execution of a service operation.

A service may have one or more service interfaces. Each of them shall be described in a separate subsection. The sub-section title shall contain the service interface name.

For each service interface, the purpose, message exchange pattern and architecture of the Interface shall be described.

A service interface supports one or several service operations. Each of them shall be described in a separate sub-section. The sub-section title shall contain the name of the operation. Each service operation sub-section shall contain the following information:

- Functionality – shall include a textual description of the operation functionality. In most situations, this will be the same as the operation description taken from suitable diagrams or API documentation.
- Parameters – shall describe the unambiguous data structure of input and output parameters of the operation (payload) by using suitable diagrams or references to existing standards and



explanatory tables as required, while avoiding duplication of documentation that already exists in other documents.

Service interfaces may include those that are not defined in an existing standard (e.g. a service may offer additional APIs to those defined in SECOM), but the use of these interfaces should not be a requirement to use the required functionality of the service. If the service is based on an existing specification (e.g. SECOM), all of the required operations in that specification must be implemented, and the guidelines in the existing specification on how unimplemented interfaces are handled must be adhered to.

It is mandatory to provide a clear description of each service operation parameter and the information about which data types defined in the service data model are used by the service operation in its input and output parameters. If such documentation exists in the referenced standards, a reference to it must be provided, but duplication of effort is not required. The clear description of parameters may also be left to the sample interface documentation if it exists. In this case, the textual description needs to highlight only limitations and constraints that the actual formal documentation cannot cover.

The service interface design should provide sample interface documentation of the service in e.g. OpenAPI format [5]. An example definition of the service instance XML (see 5.3) should also be provided to avoid duplication of work by the implementing parties.

#### **4.2.7. SERVICE DYNAMIC BEHAVIOUR**

This section describes the interactive behaviour between service interfaces (interaction specification) and, if required, between different services (orchestration or choreography).

The following types of views and diagrams can be used to describe the dynamic behaviour:

- Sequence diagrams
- Interaction diagrams
- State machine diagrams.

This section is especially relevant to extend the Service Design structure (see section 4.2.4) from a logical description of interactions to a structure where actual operations and interfaces are referenced. This will also allow a better understanding of the interaction between different services.

#### **4.2.8. REFERENCES**

The References section contains a list of all documents referred to by the Service Design (e.g., Service Specification). As a minimum, the Service Specification needs to be referenced.

## **5. SERVICE INSTANCE DESCRIPTION**

---

### **5.1. SERVICE INSTANCE DESCRIPTION**

---

The purpose of the Service Instance Description is to document in a human-readable manner all the information specific to a certain implementation and instantiation of a service. This document shall provide a detailed description of how a service is realised. In most cases, this document will be rather short, since it is expected that the implementation follows the Service Design, and it is not supposed to replicate any information from the Service Design.

Note that one service implementation may be deployed several times at different access points (thus forming several different service instances). In this case, several Service Instance Description XML files need to be



produced – one for each deployed instance, whereas the Service Instance Description can be identical (if all instances behave equivalently).

To assure a certain uniformity of Service Instance Description produced by different authors, the document shall be aligned with the Service Instance Description template ANNEX C.

## **5.2. SERVICE INSTANCE DESCRIPTION TEMPLATE**

The Service Instance Description template (see ANNEX C) shall support the service developers in creating a document-based description of the service implementation and instantiation. The template prescribes a structure of sections (to be completed by the service implementer) and for each section descriptive instructions for the intended content.

### **5.2.1. INTRODUCTION**

The introduction section contains basic information, such as the purpose of the document or the intended readership.

### **5.2.2. SERVICE INSTANCE IDENTIFICATION**

This section provides a tabular overview of mainly administrative attributes about the service instance. Example content of this section could include reference to service design, name, identifier and version of the implementation and instance, author (vendor information) and keywords.

### **5.2.3. SERVICE IMPLEMENTATION AND INSTANCE DETAILS**

This section describes any information that appears useful for the understanding of the service implementation in general and of the actual service instance in particular. This may include internal design decisions, required configuration data, deployment pre-requisites, composition of services, internal service structure, etc.

The Service Instance Description template does not prescribe a detailed format for this section.

### **5.2.4. RELEASE NOTES**

This section describes the release notes of the service instance. It shall contain at least the following set of information:

- Release identification and date.
- Feature list, including:
  - added features;
  - changed features; and
  - removed features.
- Bug list including:
  - known open bugs; and
  - resolved bugs.

The Service Instance Description template does not prescribe a detailed format for this section.

### **5.2.5. REFERENCES**

The References section contains a list of all documents referred to by the Service Instance Description (e.g., Service Specification, Service Design, etc.). As a minimum, the Service Specification and Service Design(s) need to be referenced.



### 5.3. SERVICE INSTANCE DESCRIPTION XSD STRUCTURE

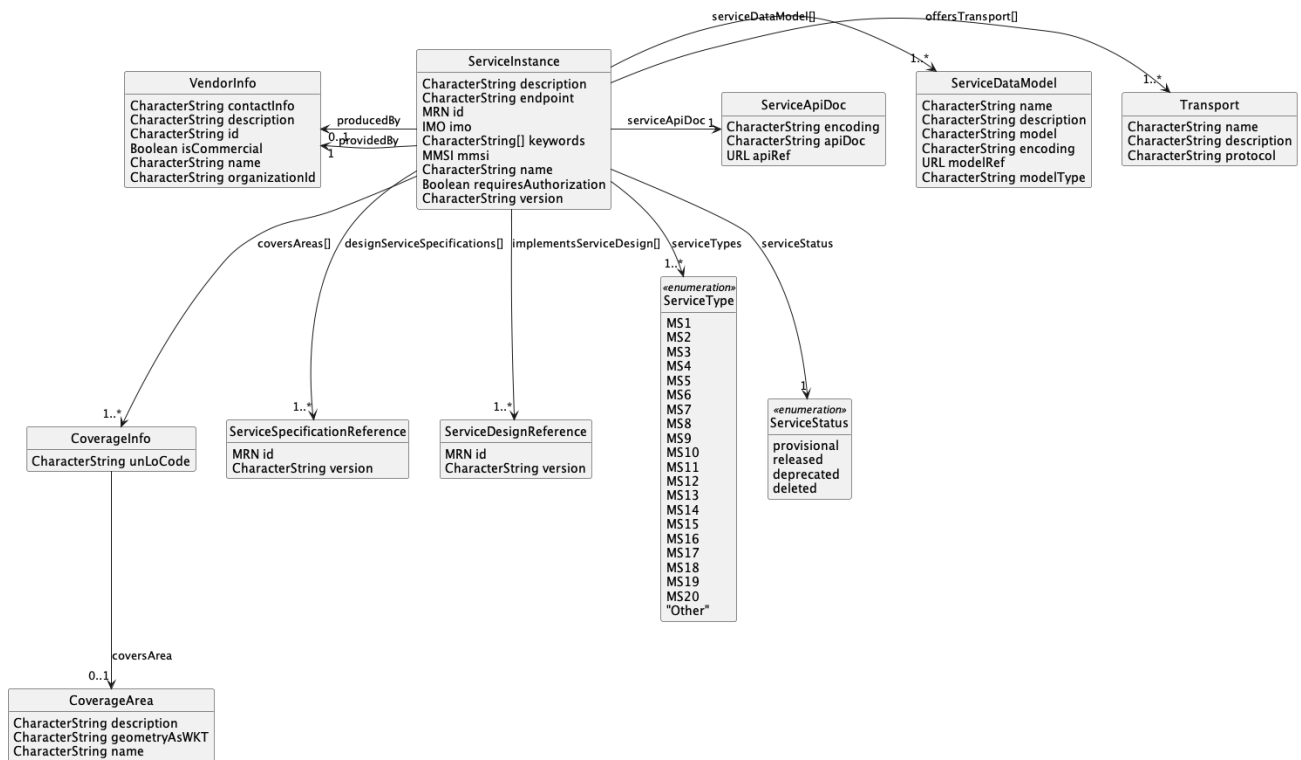


Figure 4 Structure of the Service Instance Description

Figure 4 gives an overview of the formal description of the Service Instance Description. The individual items are described in the 0.



Table 2 Information Elements of the Service Instance Description

Type Name		Description	
ServiceInstance		A Service Instance Description. One service implementation may be deployed at several places by the same or different service providers; each such deployment represents a different service instance.	
	Element Name	Type	Description
	name	CharacterString	The human-readable name of the service instance. The service name shall be at maximum a one-line brief label. Newer versions of the same Service Specification shall adopt the same name.
	id	MRN	Service identifier type to be used by Service Specifications, designs, and instances. Globally unique identification of the service instance. Newer versions of the same service instance shall adopt the same ID. The identifier must conform to a Maritime Resource Name (MRN) identity.
	version	CharacterString	Service version indicator type to be used by Service Specifications, designs, and instances. Currently, the version indicator is defined as a string. Version of the service instance. A service instance is uniquely identified by its ID and version. Any change in the Service Design reference requires a new version of the service instance.
	status	ServiceStatus <<enumeration>>	Status of the service instance. The status field has one of the following values: <ul style="list-style-type: none"><li>• provisional;</li><li>• released;</li><li>• simulated;</li><li>• deprecated; and</li><li>• deleted.</li></ul>
	description	CharacterString	A human-readable short description of the service instance. The description shall contain an abstract of what a service implementation does and what the service consumer should know about how the service implementation works.
	keywords	CharacterString	A list of keywords associated to the service. Keywords shall contain searchable words characterizing the service provided. The keyword list is recommended to reflect the type of service, but can also contain data formats, datamodels, datatypes, etc., that are possible to exchange by the service instance.
	endpoint	CharacterString	Endpoint that describes where the service endpoint is located (e.g., URL)



Type Name		Description	
	statusEndpoint	CharacterString	Mandatory endpoint that is used by the service registry to check that the service is available. Must be a URL that is accessible to the service registry.
	MMSI	CharacterString	Optional Maritime Mobile Service Identity. MMSI should be registered with 9 digits or left empty (blank).
	IMO	CharacterString	Optional International Maritime Organization (IMO) number. IMO should be registered with 7 digits or left empty (blank).
	serviceTypes	List of ServiceType <<enumeration>>	Optional field to categorize a list of servicetypes. The service type shall reflect the associated operational service type provided according to defined types in paragraph Service Type. If type not yet defined, a proposed type will be reviewed and added when accepted, i.e., an open enumeration. The following values are currently reserved and provide some room for changes in accordance with IMO Maritime Service portfolio: <ul style="list-style-type: none"><li>• MS1 – MS20</li><li>• Other.</li></ul>
	requiresAuthorization	Boolean	Indicates whether authorization is required or not.
	designServiceSpecifications	ServiceSpecificationReference	Refers to the Service Specification that the implemented Service Design is an actualization of. Must also include the version number of the Service Specification. May refer to multiple different Service Specifications and/or multiple versions of the same Service Specification.
	implementsService-Designs	ServiceDesignReference	Refers to the Service Design, including the version that is implemented by this service instance. A service may reference multiple Service Designs if it implements them. It may also refer to multiple Service Design versions if it supports them.
	coversAreas	CoverageInfo	Mandatory reference to a list of geographical areas covered by the service instance.
	producedBy	VendorInfo	An optional reference to information about the producer of the service implementation.
	providedBy	VendorInfo	Mandatory reference to information about the service provider of the service instance.



serviceApiDoc	ServiceApiDoc	<p>The API documentation of the service. This should be in OpenAPI or other widely used and standardized format. The elements of this are:</p> <ul style="list-style-type: none"><li>• encoding – the encoding of the API documentation;</li><li>• apiDoc – the actual API documentation if embedded (this is not preferred); and</li><li>• apiRef – the reference to the location of the API documentation (this is preferred).</li></ul> <p>The API doc of the implemented designs should only be modified to exclude any unsupported, unrequired operations and to combine the API docs of multiple designs if the service supports them.</p>
offersTransport	Transport	<p>Definition of the transport protocol used by the Service Design.</p> <p>Elements of a transport are:</p> <ul style="list-style-type: none"><li>• Name: Human-readable name.</li><li>• Description: Human-readable description of the transport protocol used by the Service Design.</li><li>• Protocol A non-formal string representation of the transport (e.g. http/rest, http/soap, etc.) that provides enough information to a service consumer to be able to connect.</li></ul>





serviceDataModel	List of ServiceDataModel	<p>The ServicePhysicalDataModel describes the data model for the Service Design. The ServiceDataModel describes in detail all the data structures being actually exchanged when service consumers interact with a service instance that implements this design.</p> <p>The goal is to have a reference to the actual business data model that the service uses here (e.g. S-212) while the actual interface descriptions (e.g. OpenAPI) will most likely describe the data model and parameters of the transport layer (e.g. SECOM or MMS). Thus, it is important to note that the data model needs a reference that is ideally a link to an external data model (S-100 product).</p> <p>It is important to note that for findability, having the supported data models listed in this element has value. In most cases, the actual data model will be documented in detail as a part of the API documentation:</p> <ul style="list-style-type: none"><li>• Name: Human-readable model name. The name shall be no longer than one line.</li><li>• Description: Human-readable description of the model.</li><li>• Model – the model can, e.g. be a WSDL file, a JSON API, or the like. It is recommended to wrap the model in a CDATA section and provide enough information in the name and description to make clear how to deal with the content in the model. Alternatively, the model may refer to an externally available file that defines the model URL with the following attributes. This prevents the need for duplication of effort or embedding possibly changing model documentation in this metadata file.</li><li>• modelRef – the URL of an external file that defines the data model used. An alternative to embedding the data model in the model attribute.</li><li>• modelType – the modelType shall contain an abbreviation that indicates what technology is used to describe the model. E.g. WSDL, JSON.</li><li>• Encoding – the encoding of the model.</li></ul>
Type Name	Description	
ServiceDesignReference / ServiceSpecificationReference	A reference to the Service Specification/design that is implemented by the service instance. It has the ID and the version of the respective Service Specification/design.	



Element Name		Type	Description
	id	MRN	Service identifier type to be used by Service Specifications, designs, and instances. Identification of the Service Design implemented by the service instance. The identifier must conform to a Maritime Resource Name (MRN) identity.
	version	CharacterString	Service version indicator type to be used by Service Specifications, designs, and instances. Currently, the version indicator is defined as a string. Version of the Service Design implemented by the service instance.
Type Name		Description	
CoverageInfo		Defines a geographical area from which the service instance is accessible. This is a choice between a geographical area defined by coordinates or the United Nations Code for Trade and Transport Locations (UN/LOCODE). One of the two options must be provided. Worldwide accessibility is indicated by a 'coversArea' element with a missing 'geometryAsWKT' element.	
Element Name		Type	Description
	coversArea	CoverageArea	Defines a geographical area from which the service instance is accessible.
	unLoCode	CharacterString	The "United Nations Code for Trade and Transport Locations" is commonly known as "UN/LOCODE". The unLoCode should be registered with five characters, no spaces, capital letters or left empty (blank).
Type Name		Description	
CoverageArea		Defines a geographical area from which the service instance is accessible.	
Element Name		Type	Description
	name	CharacterString	Human-readable name of the coverage area, e.g., a well-known name like 'Bermuda Triangle'. The name shall be no longer than one line.
	description	CharacterString	Human-readable description of the coverage area.
	geometryAsWKT	CharacterString	A polygon described in WKT (Well Known Text) with coordinates in a coordinate reference system EPSG:4326, e.g., POLYGON(LON1 LAT1, LON2 LAT2, LON3, LAT3, LON1 LAT1). If the element is empty, the default is the whole world.
Type Name		Description	
VendorInfo		Describes the vendor producing and/or providing the service instance.	
Element Name		Type	Description
	id	CharacterString	Unique identification of the vendor.
	name	CharacterString	Human-readable vendor name. The name shall be no longer than one line.



description	CharacterString	Human-readable description of the vendor.
contactInfo	CharacterString	Human-readable contact information of the vendor.
organizationId	CharacterString	Unique identifier of the organisation, the author belongs to. The identifier should conform to a Maritime Resource Name (MRN) identity.
isCommercial	Boolean	Optional indication on the commercial status of the vendor.

## 6. GOVERNANCE

A governance model will be needed in service management. This includes questions about the process to decide on:

- maturity of Technical Service Specifications (all artefacts);
- the scope of Technical Service Specifications;
- the evolution of Technical Service Specifications;
- the publication of Technical Service Specifications;
- the life cycle of Technical Service Specifications and service instances; and
- conformance of service instances to Technical Service Specifications.

The definition and description of governance structures and procedures are outside the scope of this document.

## 7. DEFINITIONS

The definitions of terms used in this IALA Guideline can be found in the International Dictionary of Marine Aids to Navigation (IALA Dictionary). Where conflict arises, the IALA Dictionary shall be considered as the authoritative source of definitions used in IALA documents.

In addition, for this document, the following terms are relevant:

External Data Model	Describes the semantics of the 'maritime world' (or a significant part thereof) by defining data structures and their relations. This could be at a logical level (e.g., in UML) or at the physical level (e.g., in XSD schema definitions), for example, standard data models, or S-100-based data produce specifications.
Message Exchange Pattern	<p>Describes the principles of two different parts of a message passing system (in our case: the service provider and the service consumer) interact and communicate with each other.</p> <p>Examples:</p> <p>In the Request/Response MEP, the service consumer sends a request to the service provider to obtain certain information; the service provider provides the requested information in a dedicated response.</p> <p>In the Publish/Subscribe MEP, the service consumer establishes a subscription with the service provider to obtain certain information; the service provider publishes information (either in regular intervals or upon change) to all subscribed service consumers.</p>



Operational Activity	An activity performed by an operational node. Examples of operational activities in the maritime context are Route Planning, Route Optimization, Logistics, Safety, Weather Forecast Provision, etc.
Operational Model	A structure of operational nodes and associated operational activities and their inter-relations in a process model.
Operational Node	<p>A logical entity that performs activities. Note: nodes are specified independently of any physical realization.</p> <p>Examples of operational nodes in the maritime context are: Maritime Control Centre, Maritime Authority, Ship, Port, Weather Information Provider, etc.</p>
Service	The provision of something (a non-physical object), by one, for the use of one or more others, regulated by formal definitions and mutual agreements. Services involve interactions between providers and consumers, which may be performed in a digital form (data exchanges) or through voice communication or written processes and procedures.
Service Consumer	A service consumer uses service instances provided by service providers. All users within the maritime domain can be service customers, e.g., ships and their crew, authorities, VTS stations, organizations (e.g., meteorological), commercial service providers, etc.
Service Data Model	Formal description of one dedicated service at logical level. The service data model is part of the Service Specification. Is typically defined in UML and/or XSD. If an external data model exists (e.g., a standard data model), then the service data model shall refer to it: each data item of the service data model shall be mapped to a data item defined in the external data model.
Service Design	Documents are the details of a Service Design (most likely documented by the service implementer). The Service Design description includes (but is not limited to) a service physical data model and describes the technology used, transport mechanism, quality of service, etc.
Service Implementation	The provider side implementation of a dedicated Service Design (i.e., implementation of a dedicated service in dedicated technology).
Service Implementer	Implementers of services from the service provider side and/or the service consumer side. Anybody can be a service implementer but mainly this will be commercial companies implementing solutions for shore and ship.
Service Instance	One service implementation may be deployed at several places by same or different service providers; each such deployment represents a different service instance, being accessible via different endpoints.
Service Instance Description	Documents the details of a service implementation (most likely documented by the service implementer) and deployment (most likely documented by the service provider). The Service Instance Description includes (but is not limited to) Service Design reference, service provider reference, service access information, service coverage information, etc.
Service Interface	The communication mechanism of the service, i.e., interaction mechanism between service provider and service consumer. A service interface is characterized by a message exchange pattern and consists of service operations that are either allocated to the provider or the consumer of the service.
Service Operation	Functions or procedures which enable programmatic communication with a service via a service interface.



Service Physical Data Model	<p>Describes the realization of a dedicated service data model in a dedicated technology. This includes a detailed description of the data payload to be exchanged using the chosen technology. The actual format of the service physical data model depends on the chosen technology. Examples may be WSDL and XSD files (e.g., for SOAP services) or swagger (Open API) specifications (e.g., for REST services). If an external data model exists (e.g., a standard data model), then the service physical data model shall refer to it: each data item of the service physical data model shall be mapped to a data item defined in the external data model.</p> <p>To prove the correct implementation of the Service Design, there shall exist a mapping between the service physical data model and the service data model. This means, each data item used in the service physical data model shall be mapped to a corresponding data item of the service data model. (In case of existing mappings to a common external (standard) data model from both the service data model and the service physical data model, such a mapping is implicitly given.)</p>
Service Provider	<p>A service provider provides instances of services according to a Service Design and Service Instance Description. All users within the maritime domain can be service providers, e.g., authorities, VTS stations, organizations (e.g., meteorological), commercial service providers, etc.</p>
Service Specification	<p>Describes one dedicated service at logical level. The Service Specification is technology-agnostic. The Service Specification focuses on describing the use cases of the service and high-level interactions between the service and other actors (human or services, application, etc.). The Service Specification also includes the description of the logical data model of the service with reference to any or all external data models that are used. Ideally the logical data model is a subset of an existing data model.</p>
Service Specification Producer	<p>Producers of Service Specifications in accordance with the service documentation guidelines.</p>
Service Design	<p>The Service Design of a dedicated service in a dedicated technology. One Service Specification may result in several Service Designs, realizing the service with different or same technologies.</p>

## 8. ABBREVIATIONS

---

API	Application Programming Interface
EPSG	European Petroleum Survey Group (a spatial reference system)
IMO	International Maritime Organisation
JSON	JavaScript Object Notation
LAT	Latitude
LON	Longitude
MCP	The Maritime Connectivity Platform (formerly the Maritime Cloud)
MEP	Message Exchange Pattern
MRN	Maritime Resource Name
MS	Maritime Service
REST	Representational State Transfer
SOAP	Simple Object Access Protocol
S-100	Universal Hydrographic Data Model (IHO)
UML	Unified Modelling Language
UN/LOCODE	United Nations Code for Trade and Transport Locations



URL	Uniform Resource Locator
VTs	Vessel Traffic Services
WSDL	Web Service Definition Language
XML	Extensible Mark-up Language
XSD	XML Schema Definition
YAML	YAML Ain't Markup Language

## 9. REFERENCES

---

- [1] Maritime Services (MS), NCSR 5 / WP.4 Draft Report of the Navigation Working Group
- [2] S-100 Universal Hydrographic Data Model Version 5.0.0, <https://iho.int/en/s-100-universal-hydrographic-data-model>
- [3] Maritime Resource Name (mrnregistry.org)
- [4] This concept is implemented as the Maritime Service Registry within the Maritime Connectivity Platform (MCP, formerly called the Maritime Cloud), <http://www.maritimeconnectivity.net>.
- [5] OpenAPI Initiative, <https://www.openapis.org/>
- [6] C4 Model, <https://c4model.com/>



## **ANNEX A    SERVICE SPECIFICATION TEMPLATE**

**Note:** For readability and reuse, the template file is published at the IALA website along with this guideline (search for G1128).

## **ANNEX B    SERVICE DESIGN TEMPLATE**

**Note:** For readability and reuse, the template file is published at the IALA website along with this guideline (search for G1128).

## **ANNEX C    SERVICE INSTANCE DESCRIPTION TEMPLATE**

**Note:** For readability and reuse, the template file is published at the IALA website along with this guideline (search for G1128).

## **ANNEX D    SERVICE DESIGN TEMPLATE FOR SECOM SERVICE**

**Note:** For readability and reuse, the template file is published at the IALA website along with this guideline (search for G1128).